Amendments to the Claims

1. (currently amended) In a computer system with a textual mark-up language engine, a table sizing code segment~~method~~ designed to auto-size table elements, a table cell sizing code segment~~method~~ designed to auto-size table cell elements ~~with a horizontal character flow property, a method for altering the input and output between the table sizing method and the table cell sizing method to render auto-sized textual mark-up language table cells with a vertical character flow property, the~~, an altering code segment performing a method comprising:

in the case of determining that a table sizing code segment is causing a table cell sizing code segment to execute a code path determining~~ that a call to a table cell sizing method from a table sizing method is a request to determine an acceptable~~ a minimum width table cell measurement for a given table cell element with a horizontal character flow property, ~~and the request includes~~ causing the table cell sizing code segment to execute a code path using a minimum width input designed ~~to be used by~~ for a table cell element with the horizontal character flow property~~ to render a minimum table cell width~~; and

in the case of determining that a table sizing code segment is causing a table cell sizing code segment to execute a code path determining a minimum width table cell measurement for a given table cell element with a vertical character flow property, causing the table cell sizing code segment to execute using a maximum width input that is likely to cause element content to be flowed into a single vertical line.

~~determining that the given table cell content element has the vertical character flow property;~~

~~increasing the minimum width input enough so that it is likely that all characters or objects within each paragraph in the table cell will be flowed into a separate single vertical line; and~~

~~calling the table cell sizing method with the increased input.~~

2. (currently amended) The method of claim 1, where an output from ~~a call to~~ the table cell sizing ~~method~~ code segment consists of a logical height measurement and a logical

width measurement, and the logical height measurement is returned to the table sizing code segment ~~method~~ as ~~the~~ an acceptable minimum width for the given table cell element.

3. (currently amended) The method of claim 1, where an output from ~~the call to~~ the table cell sizing ~~method~~ code segment is a measurement of a character or other object in the element content ~~element~~ with a longest logical width measurement, the longest logical width measurement being determined by measuring each object or character in the vertical direction.

4. (currently amended) The method of claim 2, wherein ~~a next call by~~ the table sizing ~~method~~ code segment ~~regarding~~ causes the table cell sizing code segment to size the given table cell element ~~is a request to size that table cell~~ at a proposed width input of less than or equal to said logical height returned to the table sizing ~~method~~ code segment as the acceptable minimum table cell width plus some minimal acceptable error, and an output of the table cell sizing code segment using the proposed width input comprises a new ~~the~~ logical height and a new logical width measurement~~s~~, which are returned to the table sizing ~~method~~code segment, the new logical height returned as physical width, and the new logical width returned as physical height.

5. (currently amended) The method of claim 4, wherein the minimal acceptable ~~minimum~~ error ranges from ten to thirty percent of the proposed width.

6. (currently amended) In a computer system with a textual mark-up language engine, a table sizing ~~method~~ code segment designed to auto-size table elements, a table cell sizing ~~method~~ code segment designed to auto-size table cell elements with a horizontal character flow property, an altering ~~method~~ code segment for altering communications between the table sizing ~~method~~ code segment and the table cell sizing ~~method~~ code segment to render auto-sized textual mark-up language table cells with a vertical character flow property, the altering code segment performing a method comprising:

determining that a ~~call to a~~ table ~~cell~~ sizing code segment ~~method from~~ is attempting to cause a table cell sizing code segment ~~method~~ ~~is a request~~ to execute a code path that ~~to~~ determines an acceptable maximum width table cell measurement for a given table cell element using ~~and the request includes~~ a maximum width input designed to be used by the table cell sizing code segment to measure a table cell element with the horizontal character flow property to render a maximum table cell horizontal width;

determining that the given table cell ~~content~~ element has ~~the~~ a vertical character flow property;

determining an average character logical width for a language contained in the given table cell element;

determining a maximum distance that characters will be allowed to flow in the vertical character flow direction, the maximum distance determination being made based on some empirically determined number N multiplied by the average character logical width; and

causing ~~calling~~ the table cell sizing ~~method~~ code segment to execute a code path using the determined maximum distance as input instead of ~~with~~ the maximum width input ~~changed to the determined distance that characters will be allowed to flow in the character flow direction~~.

7. (currently amended) The method of claim 6, where an output from the ~~call to the~~ table cell sizing code segment ~~method~~ consists of logical height and logical width measurements and the logical height measurement is returned to the table sizing ~~method~~ code segment as the acceptable maximum width table cell.

8. (original) The method of claim 6, where the empirically determined number N is empirically determined based on the average number of characters in a vertical line in print or other popular media that typically displays that character set.

9. (original) The method of claim 6, where said determining a maximum distance that characters will be allowed to flow in the character flow direction is increased to the greater of the longest object logical width or the empirically determined number N multiplied by the average character logical width.

10. (currently amended) The method of claim 6, where ~~a next call by~~ the table sizing ~~method~~ code segment causes ~~regarding~~ the table cell sizing code segment to size the given table cell element ~~is a request to size that table cell~~ at a proposed physical width input greater than or

equal to the logical height returned to the table sizing ~~method~~ code segment as the maximum acceptable width table cell, and an output of the table cell sizing code segment using the proposed physical width input comprises a new ~~the~~ logical height and a new logical width measurement~~s~~, which are returned to the table cell sizing code segment ~~method~~, the new logical height returned as physical width, and the new logical width returned as physical height.

11. (currently amended) A method for auto-sizing textual mark-up language table cells with a vertical character flow property, the method comprising:

determining that a table cell has the vertical character flow property;

estimating a logical width for input to a table cell sizing method designed to auto-size a table cell with the horizontal character flow property; and

calling the table cell sizing method with the estimated logical width input in order to auto-size the table cell with the vertical character flow property;

wherein the estimated logical width is determined by,

dividing the area of a minimum logical height rectangle by a physical width proposed by a table sizing method,

dividing the area of a maximum logical height rectangle by a physical width proposed by a table sizing method, or

summing the areas of a minimum logical height rectangle with a maximum logical height rectangle, and dividing that sum by twice a physical width proposed by a table sizing method.

12. (canceled) ~~The method of claim 11 where the estimated logical width input is determined by dividing the area of a minimum logical height rectangle by a physical width proposed by a table sizing method.~~

13. (canceled) ~~The method of claim 11 where the estimated logical width input is determined by dividing the area of a maximum logical height rectangle by a physical width proposed by a table sizing method.~~

14. (canceled) ~~The method of claim 11 where the estimated logical width input is determined by averaging the logical heights of a maximum logical height rectangle and a minimum logical height rectangle.~~

15. (canceled) ~~The method of claim 11 where the estimated logical width input is determined by summing the areas of a minimum logical height rectangle with a maximum logical height rectangle, and dividing that sum by twice a physical width proposed by the table sizing method.~~

16. (original) The method of claim 11, where an output from a call to the table cell sizing method is examined and a determination is made as to whether a resulting logical height measurement output is within a minimum acceptable error of a physical width proposed by the table sizing method.

17. (original) The method of claim 11, where if a returned actual logical height output is within an acceptable minimum error of a physical width proposed by the table sizing method, the actual logical width output is returned to the table sizing method as a physical height, and a returned logical height output is returned to the table sizing method as a physical width.

18. (currently amended) In a computer system with a textual mark-up language engine, a parent textual mark-up language element with a horizontal character flow property, a child textual mark-up language element with a vertical character flow property, a method for determining a proposed logical width dimension, the method being given a proposed physical width, a desired layout area, and the child element, the method comprising:

determining an area of a minimum logical height rectangle for the child element; ~~by flowing each paragraph or sentence in the child element into a single line in the character flow direction; and~~

~~obtaining an actual logical width of a minimum logical height rectangle by measuring a length of a longest line in the child element in the character flow direction;~~

~~obtaining an actual logical height of a minimum logical height rectangle by finding and summing the measurements of each character or object in each said line that that occupies the most space in the direction perpendicular to the character flow direction and adding a minimal space between lines to improve readability; and~~

~~multiplying the obtained actual logical height by the obtained actual logical width;~~

determining an area of a maximum logical height rectangle for the child element; ~~by;~~

~~setting a maximum logical width for the maximum logical height rectangle;~~

~~flowing the child element into the maximum logical height rectangle in a line in the character flow direction so long as no next character or object in the content element being flowed into the line would cause the length of the line to exceed the set maximum logical width;~~

~~starting a new line next to and parallel to the previous line and flowing the content element into that line each time the next character or object in the child element would cause the line then being flowed into the rectangular area to exceed the set maximum logical width;~~

~~obtaining an actual logical width of the maximum logical height rectangle by finding and measuring a length of the longest line in the character flow direction;~~

~~obtaining an actual logical height of the minimum logical height rectangle by finding and summing the measurements of each character or object in each said line that occupies the most space in the direction perpendicular to the character flow direction and adding a minimal space between lines to improve readability;~~

~~multiplying the obtained actual logical height by the obtained actual logical width in order to determine the area of the maximum logical height rectangle;~~

determining a proposed logical width by summing the areas of the determined maximum logical height rectangle and the determined minimum logical height rectangle and dividing the sum by approximately twice the proposed physical width.

19. (original) The method of claim 18, where the determined proposed logical width is set at the greater of itself or the value determined by dividing the area of the determined minimum logical height rectangle area by the proposed physical width.

20. (original) The method of claim 18, where the determined proposed logical width value is altered, the method comprising:

determining an average character logical width value for a language character set by accessing a table containing a value for the language character set;

obtaining a subtracted value by subtracting the integer value one from the determined proposed logical width value;

obtaining a divided value by dividing the subtracted value by the determined average character logical width value;

obtaining an added value by adding the integer value one to the divided value; and

obtaining a new proposed logical width by multiplying the added value by the determined average character logical width value.

21. (original) A computer readable medium having stored thereon instructions operational to auto-size a vertical character flow textual mark-up language element within an area of an output device, the instructions comprising:

a first program module designed to accept a request to auto-size textual mark-up language elements;

a second program module designed to accept requests to auto-size a textual mark-up language element with the horizontal character flow property at a requested horizontal width;

a third program module designed to alter the request made to the second program module by the first program module so that while the second program module continues to operate as it would for a textual mark-up language element with the horizontal character flow property, the second program module is actually auto-sizing a textual mark-up element with the vertical character flow property and the resulting auto-sized textual mark-up language element is returned to the first program module within the rectangular area within an acceptable error of the requested horizontal width.

22. (previously presented) A computer readable medium having stored thereon instructions operational to auto-size a textual mark-up language table element within an area of an output device, a first program module designed to auto-size a textual mark-up language table

element, a second program module designed to accept a request from the first program module to determine the minimum and maximum table cell element width measurements, a third program module that affects the results of the request made by the first program module to the second program module whenever the request to determine the minimum and maximum table cell element width measurement is for a table cell element with the vertical character flow property, the instructions comprising:

instructions affecting the results of the minimum table cell element width measurement by altering the input to the second program module so that characters are allowed to flow in the character flow direction until all characters in each paragraph flow into a single line in the character flow direction;

instructions affecting the results of the maximum table cell element width measurement by altering the input to the second program module so that each time characters flowed into a line in the character flow direction exceed approximately N characters, a new line in the character flow direction is started immediately next to and parallel to the previous line;

wherein the first module is designed to request auto-sizing of table cell elements with the horizontal character flow property and the third module alters inputs of the first module's requests to the second module when the request is to auto-size a table cell element with the vertical character flow property.

23. (original) The computer readable medium of claim 22, wherein the instructions of the second program module are operational to determine the dimensions occupied by a table cell element, the instructions comprising:

determining a minimum width table cell element's height and width dimensions by measuring the table cell element after the contents of each paragraph within the table cell element are flowed into a single line in the character flow direction;

determining a maximum width table cell element's height and width dimensions by measuring the table cell element after the contents of the table cell element are flowed into a line in the character flow direction and when each line exceeds approximately N characters, a new line in the character flow direction is started immediately next to and parallel to the previous line; and returning the maximum and minimum width table cell element's height and width dimensions to the third program module.

24. (original) The computer readable medium of claim 23, wherein the instructions of the third program module are operational to change the logical perspective of the maximum and minimum table cell element height and width dimensions, the instructions comprising:

obtaining from the second program module the maximum width table cell element's height and width dimensions and returning the height dimension to the first program module as the maximum table cell width; and

obtaining from the second program module the minimum width table cell element's height and width dimensions and returning the height dimension to the first program module as the minimum table cell width.

25. (previously presented) A computer readable medium having stored thereon instructions operational to auto-size a textual mark-up language table element on an output device, a first program module designed to auto-size a textual mark-up language table element, a second program module designed to accept a request from the first program module to auto-size a textual mark-up language table cell element at a proposed width, a third program module that affects the results of the request made by the first program module to the second program module whenever the request to auto-size a textual mark-up language table cell element at a proposed width is for a table cell element with a_vertical character flow property, the instructions comprising:

determining that a table cell element has the vertical character flow property;

estimating a distance to allow characters to flow in the character flow direction, so that when the contents of the table cell element are flowed into the table cell element at the estimated distance, the resulting distance perpendicular to the character flow direction is within an acceptable error of the proposed width;

wherein the second module is designed to receive requests to auto-size table cell elements with a horizontal character flow property and the third module alters an input of a request from the first module to the second module when the request is to auto-size a table cell element with the vertical character flow property.

26. (original) The computer readable medium of claim 25, wherein the second program module flows the table cell element at the estimated distance in the character flow direction, and the third program module returns the flowed table cell element to the first program module rotated ninety degrees clockwise.

27. (original) The computer readable medium of claim 25, wherein the instructions of the third program module are operational in estimating the distance to allow characters to flow in the character flow direction for the table cell, the instructions comprising:

summing the areas of a minimum width table cell element and a maximum width table cell element and dividing the sum by twice the proposed width.

28. (original) A system for rendering textual mark-up language elements with the vertical character flow property on an output device, the system comprising:

a first object designed to accept a request to auto-size textual mark-up language elements;

a second object designed to accept requests to auto-size a textual mark-up language element with the horizontal character flow property at a defined horizontal width; and

a third object designed to alter the request made to the second object by the first object so that while the second object continues to operate as it would for a textual mark-up language element with the horizontal character flow property, the second object is actually auto-sizing a textual mark-up element with the vertical character flow property and the resulting auto-sized element is returned to the first object rotated ninety degrees clockwise.

29. (original) A system for rendering textual mark-up language elements with the vertical character flow property on an output device, a first object designed to auto-size a textual mark-up language table element, a second object designed to accept a request from the first object to determine a minimum and maximum table cell element width measurement, a third object that affects the results of the request made by the first object to the second object whenever the request to determine the minimum and maximum table cell element width

measurements is for a table cell element with the vertical character flow property, the system comprising:

the third object altering the minimum table cell width request to the second object so that characters are allowed to flow in the character flow direction until all characters in each paragraph flow into a single line in the character flow direction, and the second object returns the resulting height and width measurements of the table cell element to the third object, the returned width measured in the direction parallel to the character flow direction, and the returned height measured in the direction perpendicular to the character flow direction, and the third object forwards the returned height measurement to the first object as the measurement for the minimum table cell width; and

the third object altering the maximum table cell width request to the second object so that each time characters flowed into a line in the character flow direction exceeds approximately N characters, a new line in the character flow direction is started immediately next to and parallel to the previous line, and the second object returns the resulting height and width measurements of the table cell element to the third object, the returned width measured in the direction parallel to the character flow direction, and the returned height measured in the direction perpendicular to the character flow direction, and the third object forwards the returned height measurement to the first object as the measurement for the maximum table cell width.

30. (original) The system of claim 29, wherein based on the minimum and maximum table cell widths forwarded to the first object by the third object for the table cell element, the first object later requests auto-sizing of the table cell element at a proposed width, the system comprising:

the third object estimating a distance in the character flow direction to flow the contents of the table cell element so that the resulting table cell element measurement in the direction perpendicular to the character flow direction is within an acceptable error of the proposed width;

the second object then flows the contents of the table cell element in the character flow direction at the estimated distance and then measures the table cell element dimensions;

the third object then compares the flowed table cell element dimensions to the requested width, and if the table cell element measurement in the direction perpendicular to the character flow direction is within an acceptable error of the proposed width, then the rendered table cell element is forwarded to the first object rotated ninety degrees clockwise.

31. (original) The system of claim 30, wherein the third object re-estimates a distance to flow the characters in the character flow direction followed by the second object re-flowing the contents of the table cell at the re-estimated distance and returning the table cell dimensions to the third object until the third object determines that the table cell measurement in the direction perpendicular to the character flow direction is within an acceptable error of the proposed width, in which case the third object forwards the rendered table cell to the first object rotated ninety degrees clockwise.

32. (original) The system of claim 30, wherein the third object estimating a distance in the character flow direction is estimated by summing the areas of the minimum and maximum width table cells for the table cell being measured and dividing the sum of the areas by twice the proposed width.

33. (new) The method of claim 18, wherein determining an area of a minimum logical height rectangle for the child element comprises:

flowing each paragraph or sentence in the child element into a single line in the character flow direction;

obtaining an actual logical width of a minimum logical height rectangle by measuring a length of a longest line in the child element in the character flow direction;

obtaining an actual logical height of a minimum logical height rectangle by finding and summing the measurements of each character or object in each said line that that occupies the most space in the direction perpendicular to the character flow direction and adding a minimal space between lines to improve readability; and

multiplying the obtained actual logical height by the obtained actual logical width.

34. (new) The method of claim 18, wherein determining an area of a maximum logical height rectangle for the child element comprises:

setting a maximum logical width for the maximum logical height rectangle;

flowing the child element into the maximum logical height rectangle in a line in the character flow direction so long as no next character or object in the content element being flowed into the line would cause the length of the line to exceed the set maximum logical width;

starting a new line next to and parallel to the previous line and flowing the content element into that line each time the next character or object in the child element would cause the line then being flowed into the rectangular area to exceed the set maximum logical width;

obtaining an actual logical width of the maximum logical height rectangle by finding and measuring a length of the longest line in the character flow direction;

obtaining an actual logical height of the minimum logical height rectangle by finding and summing the measurements of each character or object in each said line that occupies the most space in the direction perpendicular to the character flow direction and adding a minimal space between lines to improve readability; and

multiplying the obtained actual logical height by the obtained actual logical width in order to determine the area of the maximum logical height rectangle.